Amendments to the Claims:

This listing of claims replaces all prior versions and listings of claims in the application:

Listing of Claims:

1. (Original) A method for embedding a foreign data block in a host data file, comprising:

receiving a host data file, the host data file having a host data file format;

receiving a foreign data block;

determining characteristics of the foreign data block;

generating packing data that describes the characteristics of the foreign data block, including data marking the beginning and end of the foreign data block and further including an identifier designed to be distinguishable from all other data in the host data file; and

embedding the packing data and the foreign data block as a foreign data block packet in the host data file;

whereby the foreign data block is identifiable, extractable, and modifiable by computer programs not configured to recognize the host file data format.

2. (Original) The method of claim 1, wherein:

generating packing data includes generating a header for the foreign data block, the header including the identifier and indicating the beginning of the foreign data block packet and the beginning of the foreign data block.

3. (Original) The method of claim 2, wherein:

generating a header includes generating a header that indicates the end of the foreign data block packet.

4. (Original) The method of claim 2, wherein:

generating packing data includes generating a trailer for the foreign data block, the trailer indicating the end of the foreign data block.

5. (Original) The method of claim 2, wherein:

generating a header includes generating a header that indicates the end of the foreign data block.

6. (Original) The method of claim 1, further comprising:

including padding in the foreign data block packet to allow in-place modifications of the foreign data block that cause the foreign data block to expand.

7. (Original) The method of claim 6, wherein:

determining characteristics of the foreign data block includes determining a size of the foreign data block; and

the amount of padding is a function of the size of the foreign data block.

8. (Original) The method of claim 1, wherein:

the foreign data block is a data block not native to the host file format.

9. (Original) The method of claim 1, wherein the foreign data block is an Extensible Markup Language (XML) document and the host file is in a non-XML format.

10. (Currently Amended) The method of claim 9, wherein:

determining characteristics of the foreign data block includes determining a byte order and an encoding format of the foreign data block; and

generating ~~a header~~ packing data includes generating a header that includes information for specifying the byte order and encoding format of the foreign data block, the encoding format being one of an 8, 16, or 32 bit Unicode format.

11. (Original) The method of claim 1, wherein generating an identifier includes generating a different identifier for each different type of foreign data block when there are multiple types of foreign data blocks in the host data file.

12. (Original) The method of claim 1, wherein the foreign data block includes metadata information that describes the host data file.

13. (Original) The method of claim 12, wherein:

receiving a host data file includes receiving a host data file having a non-XML format.

14. (Original) A computer program product, tangibly stored on a machine-readable medium, comprising instructions operable to cause a programmable processor to:

receive a host data file, the host data file having a host data file format that is not understood by the computer program product;

scan for a header that indicates the beginning of an embedded foreign data block packet that contains a foreign data block, the foreign data block having a format that is recognizable by the computer program, the header including an identifier designed to be distinguishable from all other data in the host data file, the header further describing the characteristics of the foreign data block; and

process the header when the header is located.

15. (Original) The computer program product of claim 14, further comprising instructions to:

process the foreign data block.

16. (Original) The computer program product of claim 15, further comprising instructions to:

stop processing the foreign data block when a trailer is detected, wherein the trailer indicates the end of the foreign data block.

17. (Original) The computer program product of claim 16, further comprising instructions to:

modify the foreign data block as specified by a user;

ensure that the modified foreign data block fits in the foreign data block packet; and

re-embed the modified foreign data block in place of the original foreign data block.

18. (Original) The computer program product of claim 16, further comprising instructions to:

modify the foreign data block as specified by a user;

rewrite the foreign data block packet;

ensure that the re-written foreign data block packet is the same size as the original foreign data block packet; and

re-embed the re-written foreign data block packet in place of the original foreign data block packet.

19. (Original) A method for embedding a foreign data block in a host data file, comprising:

receiving a host data file, the host data file having a host data format;

receiving a packet to be embedded into the host data file, the packet including a foreign data block and further including a header and a trailer that delimit the foreign data block, the header including an identifier designed to be distinguishable from all other data in the host data file; and

embedding the packet in the host data file;

whereby the foreign data block is identifiable, extractable, and modifiable by computer programs not configured to recognize the host file data format.

20. (Original) A computer program product, tangibly stored on a machine-readable medium, for embedding a foreign data block in a host data file, comprising instructions operable to cause a programmable processor to:

receive a host data file, the host data file having a host data format that is a native file format for the computer program product;

receive a packet to be embedded into the host data file, the packet including a foreign data block that is not native to the host data file format and further including a header and a trailer that delimit the foreign data block, the header including an identifier that is designed to be distinguishable from all other data in the host data file; and

embed the packet in the host data file;

whereby the foreign data block is identifiable, extractable, and modifiable by computer programs not configured to recognize the host file data format.

21. (Original) A computer program product, tangibly stored on a machine-readable medium, for embedding a foreign data block in a host data file, comprising instructions operable to cause a programmable processor to:

receive a host data file, the host data file having a host data file format that is a native file format for a host application;

receive a foreign data block, the foreign data block being a data block that is not native to the host data file format;

determine characteristics of the foreign data block;

generate information that describes the characteristics of the foreign data block, including information marking the beginning and end of the foreign data block and further including an identifier designed to be distinguishable from all other data in the host data file; and

embed the information and the foreign data block as a foreign data block packet in the host data file;

whereby the foreign data block is identifiable, extractable, and modifiable by computer programs not configured to recognize the host file data format.

22. (Original) A computer program product, tangibly stored on a machine-readable medium, for embedding metadata in a host data file having a non-XML format, comprising instructions operable to cause a programmable processor to:

receive a host data file having a format that is not XML and that is a native file format for a host application;

receive metadata having a format that is not native to the host data file format;

determine characteristics of the metadata;

generate information that describes the characteristics of the metadata, including information marking the beginning and end of the metadata and further including an identifier designed to be distinguishable from all other data in the host data file; and

embed the information and the metadata as a packet in the host data file;

whereby the metadata is identifiable, extractable, and modifiable by computer programs not configured to recognize the host file data format.

23. (New) The product of claim 21, further comprising instructions to:

generate a header for the foreign data block, the header including the identifier and indicating the beginning of the foreign data block packet and the beginning of the foreign data block.

24. (New) The product of claim 21, further comprising instructions to:

generate a trailer for the foreign data block, the trailer indicating the end of the foreign data block.

25. (New) The product of claim 21, further comprising instructions to:

include padding in the foreign data block packet to allow in-place modifications of the foreign data block that cause the foreign data block to expand.

26. (New) The product of claim 21, wherein the foreign data block is an Extensible Markup Language (XML) document and the host file is in a non-XML format.

27. (New) The product of claim 26, further comprising instructions to:

determine a byte order and an encoding format of the foreign data block; and

generate a header that includes information for specifying the byte order and encoding format of the foreign data block, the encoding format being one of an 8, 16, or 32 bit Unicode format.

28. (New) The product of claim 21, further comprising instructions to:

generate a different identifier for each different type of foreign data block when there are multiple types of foreign data blocks in the host data file.